



*Federação das Indústrias do Estado da Bahia*

**CENTRO UNIVERSITÁRIO SENAI CIMATEC**

**Engenharia Elétrica**

**Projeto Theoprax de Conclusão de Curso**

**Desenvolvimento do robô de inspeção.**

Apresentada por: Michael Faraday  
John Nash  
James Clerk Maxwell  
Nikola Tesla

Orientador: Prof. Marco Reis, M.Eng.

Setembro de 2018

Michael Faraday  
John Nash  
James Clerk Maxwell  
Nikola Tesla

## **Desenvolvimento do robô de inspeção.**

Projeto Theoprax de Conclusão de Curso apresentada ao , Curso de Engenharia Elétrica do Centro Universitário SENAI CIMATEC, como requisito parcial para a obtenção do título de **Bacharel em Engenharia.**

Área de conhecimento: Interdisciplinar

Orientador: Prof. Marco Reis, M.Eng.

Salvador  
Centro Universitário SENAI CIMATEC  
2016

Dedico este trabalho a ...

---

## Agradecimentos

---

Salvador, Brasil  
dia de Setembro de 2018

Michael Faraday  
John Nash  
James Clerk Maxwell  
Nikola Tesla

---

## Resumo

---

Escreva aqui o resumo da dissertação, incluindo os contextos geral e específico, dentro dos quais a pesquisa foi realizada, o objetivo da pesquisa, assunção filosófica, os métodos de pesquisa usados e as possíveis contribuições que o que é proposto pode trazer à sociedade.

**Palavras-chave:** Palavra-chave 1, Palavra-chave 2, Palavra-chave 3, Palavra-chave 4, Palavra-chave 5

---

## Abstract

---

Escreva aqui, em inglês, o resumo da dissertação, incluindo os contextos geral e específico, dentro dos quais a pesquisa foi realizada, o objetivo da pesquisa, assunção filosófica, os métodos de pesquisa usados e as possíveis contribuições que o que é proposto pode trazer à sociedade.

**Keywords:** Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5

---

# Sumário

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	3
1.1.1	Objetivos Específicos . . . . .	3
1.2	Justificativa . . . . .	4
1.3	Requisitos do cliente . . . . .	4
1.4	Organização do Projeto Theoprax de Conclusão de Curso . . . . .	4
<b>2</b>	<b>Conceito do Sistema</b>	<b>5</b>
2.1	Estudo do estado da arte . . . . .	5
2.2	Descrição do sistema . . . . .	5
2.2.1	Especificação técnica . . . . .	6
2.2.2	Arquitetura geral do sistema . . . . .	6
2.2.3	Arquitetura de software . . . . .	6
2.2.3.1	Driver Layer . . . . .	7
2.2.3.2	Business Layer . . . . .	8
2.2.3.3	User Interface Layer . . . . .	8
2.3	Desdobramento da função qualidade . . . . .	9
2.3.1	Requisitos técnicos . . . . .	9
<b>3</b>	<b>Materiais e Métodos</b>	<b>10</b>
3.1	Especificação dos componentes . . . . .	10
3.1.1	Estrutura analítica do protótipo . . . . .	10
3.1.2	Lista de componentes . . . . .	10
3.2	Diagramas mecânicos . . . . .	10
3.3	Modelo esquemático de alimentação e comunicação . . . . .	10
3.3.1	Diagramas elétricos . . . . .	10
3.3.2	Esquemas eletrônicos . . . . .	11
3.4	Especificação das funcionalidades . . . . .	11
3.4.1	Fluxo das informações . . . . .	11
3.4.2	Motion Planning . . . . .	11
3.4.2.1	Definição da funcionalidade . . . . .	11
3.4.2.2	Dependências . . . . .	11
3.4.2.3	Premissas Necessárias . . . . .	11
3.4.2.4	Descrição da Funcionalidade . . . . .	12
3.4.2.5	Saídas . . . . .	13
3.4.3	Actuation . . . . .	14
3.4.3.1	Definição da funcionalidade . . . . .	14
3.4.3.2	Dependências . . . . .	14
3.4.3.3	Premissas Necessárias . . . . .	14
3.4.3.4	Descrição da Funcionalidade . . . . .	14
3.4.3.5	Saídas . . . . .	15
3.4.4	Power Management . . . . .	16
3.4.4.1	Definição da funcionalidade . . . . .	16
3.4.4.2	Dependências . . . . .	16
3.4.4.3	Premissas Necessárias . . . . .	16

---

3.4.4.4	Descrição da Funcionalidade . . . . .	16
3.4.4.5	Saídas . . . . .	17
3.4.5	System Integrity Check . . . . .	18
3.4.5.1	Definição da funcionalidade . . . . .	18
3.4.5.2	Dependências . . . . .	18
3.4.5.3	Premissas Necessárias . . . . .	18
3.4.5.4	Descrição da Funcionalidade . . . . .	19
3.4.5.5	Saídas . . . . .	20
3.5	Interface do Usuário . . . . .	20
3.6	Simulação do sistema . . . . .	20
<b>4</b>	<b>Resultados</b>	<b>21</b>
4.1	Testes unitários . . . . .	21
4.2	Testes integrados . . . . .	21
4.3	Avaliação da prontidão tecnológica . . . . .	21
4.4	Trabalhos futuros . . . . .	21
<b>5</b>	<b>Conclusão</b>	<b>22</b>
5.1	Considerações finais . . . . .	22
<b>A</b>	<b>QFD</b>	<b>23</b>
<b>B</b>	<b>Diagramas mecânicos</b>	<b>24</b>
<b>C</b>	<b>Diagramas eletro-eletrônicos</b>	<b>25</b>
<b>D</b>	<b>Wireframes</b>	<b>26</b>
<b>E</b>	<b>Logbook</b>	<b>27</b>
	<b>Referências</b>	<b>28</b>



---

## Lista de Tabelas

---

---

## Lista de Figuras

---

1.1	Inspeção de linhas de transmissão feita por aeronaves tripuladas. . . . .	2
1.2	Interação humana durante a inspeção de linhas de transmissão. . . . .	2
1.3	Realização de inspeção em linhas de transmissão através da observação humana. . . . .	3
2.1	Arquitetura Geral da Perception . . . . .	7
3.1	Fluxograma de funcionamento da funcionalidade de Motion Planning . . .	13
3.2	Fluxograma da funcionalidade Actuation . . . . .	15
3.3	Fluxograma de funcionamento da funcionalidade de Power Management . .	17
3.4	Fluxograma da rotina para checagem do sistema . . . . .	19

---

## Lista de Siglas

---

THEOPRAX

WWW ..... World Wide Web

---

## Lista de Simbolos

---

$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble
$\partial$ .....	Bla bla bla
$\Pi$ .....	ble ble ble

---

## Introdução

---

No Brasil, a eletricidade é gerada por hidrelétricas, termoeletricas, parques eólicos e usinas nucleares. Na maioria dos casos, devido a condições geográficas e de segurança, a energia gerada nem sempre é utilizada ou consumida no local de sua geração. Portanto, há a necessidade do uso de linhas de transmissão para transportar energia gerada na fonte geradora para a carga do consumidor (??). O mercado consumidor brasileiro é composto de cerca de 47 milhões de unidades. Em termos de linhas de transmissão de energia, são cerca de 98.648,3 km, que devem estar operando 24 horas por dia, 7 dias por semana, 365 dias por ano e em perfeito estado de manutenção, para garantir eletricidade para os consumidores (??)

No Brasil, há uma quantidade considerável de linhas de transmissão de alta tensão que já ultrapassaram a vida útil as quais foram destinadas. Com o envelhecimento dos cabos, a inspeção para manutenção preventiva é um fator de extrema relevância para garantir o perfeito funcionamento dos sistemas elétricos. De um modo geral, as inspeções nas linhas de transmissão de alta tensão são realizadas regularmente de forma visual, a fim de identificar a necessidade da realização de manutenções preventivas. As inspeções buscam verificar a integridade física dos componentes das linhas, em termos de fissuras, corrosão e eventuais danos que venham a prejudicar o fornecimento de energia elétrica. Essas inspeções envolvem a análise da integridade estrutural das torres, da condição dos isoladores, das conexões das linhas de transmissão, dentre outros, a fim de se verificar a existência de eventuais pontos de ruptura.

Um dos métodos empregados para detecção de pontos quentes nos cabos é o imageamento térmico, que é capaz de identificar uma elevação de temperatura nos cabos, o que é um indício de possíveis pontos de ruptura. A inspeção através de câmera térmica é uma importante ferramenta no campo das inspeções para manutenções preventivas. Outros pontos a serem inspecionados envolvem as condições do local onde as torres são instaladas, pois a vegetação e eventuais construções devem ser mantidas a uma distância mínima segura, tal que não ocorra nenhum contato entre quaisquer estruturas e as torres ou cabos de transmissão, evitando assim interferências no funcionamento da linha.

Além disso, é essencial a garantia de dispor-se de um terreno em condições de trânsito de veículos para o transporte do pessoal de manutenção, transporte de ferramentas, dentre outros fatores. Durante vários anos, a inspeção de linhas de transmissão de alta tensão tem sido feita regularmente através de aeronaves tripuladas. As aeronaves executam vôos

em baixa altitude e muito próximos das linhas de transmissão conforme mostrado nas Figuras 1.1 e 1.2.



Figura 1.1: Inspeção de linhas de transmissão feita por aeronaves tripuladas.



Figura 1.2: Interação humana durante a inspeção de linhas de transmissão.

Em alguns casos, devido às características geográficas da região, condições climáticas e outros fatores que venham a dificultar o sobrevôo, há uma grande exposição dos tripulantes a riscos associados à tarefa. Além dos perigos aos quais os tripulantes são expostos, a inspeção feita com aeronaves tem um custo bastante elevado. Outra forma alternativa de inspeção é o uso de veículos terrestres, porém essa forma é muito limitada, pois boa parte das linhas de transmissão está localizada em áreas de difícil acesso terrestre, muitas vezes restritas pelas características geográficas da região. Além disso, o ângulo de visão é, muitas vezes, desfavorável para a realização da inspeção.

Outra maneira de inspecionar as linhas de transmissão é através de eletricitistas que literalmente caminham sobre os cabos de linhas de transmissão de alta tensão (Figura 1.3),



Figura 1.3: Realização de inspeção em linhas de transmissão através da observação humana.

realizando inspeção visual e termográfica. Esse tipo de inspeção é lenta e não é viável, tendo em vista que o país possui milhares de quilômetros de linhas de transmissão.

Neste contexto vários robôs de inspeção de linhas de transmissão foram desenvolvidos, porém poucos deles consistiram em projetos de engenharia que sejam aplicáveis no mundo real, além disso a maioria eram robôs tele-operados, ou seja robôs controlados por seres humanos. Um dos pontos diferenciais deste projeto de tese é a proposição de um desenvolvimento de uma navegação autônoma utilizando técnicas de aprendizagem de máquinas até então não utilizadas em robôs de inspeção de linhas de transmissão de alta tensão.

## **1.1 Objetivos**

veja

Nesta seção os objetivos principal (também pode-se utilizar a palavra meta) da monografia de graduação ou especialização, dissertação de mestrado ou tese de doutorado são apresentados.

### **1.1.1 Objetivos Específicos**

Nesta seção os objetivos específicos (também pode-se utilizar a palavra meta) da monografia de graduação ou especialização, dissertação de mestrado ou tese de doutorado são apresentados.

## ***1.2 Justificativa***

O pesquisador/estudante deve apresentar os aspectos mais relevantes da pesquisa ressaltando os impactos (e.g. científico, tecnológico, econômico, social e ambiental) que a pesquisa causará. Deve-se ter cuidado com a ingenuidade no momento em que os argumentos forem apresentados.

## ***1.3 Requisitos do cliente***

asjdfllkasjdlfjsdlk;f

## ***1.4 Organização do Projeto Theoprax de Conclusão de Curso***

Este documento apresenta  $x$  capítulos e está estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** Contextualiza o âmbito, no qual a pesquisa proposta está inserida. Apresenta, portanto, a definição do problema, objetivos e justificativas da pesquisa e como este projeto theoprax de conclusão de curso está estruturado;
- **Capítulo 2 - Nome do capítulo:** XXX;
- **Capítulo 5 - Conclusão:** Apresenta as conclusões, contribuições e algumas sugestões de atividades de pesquisa a serem desenvolvidas no futuro.



---

## Conceito do Sistema

---

A percepção é, de acordo com o dicionário [Michaelis \(2004\)](#), a capacidade de distinguir por meio dos sentidos ou da mente.

Segundo [Thorpe et al. \(2003\)](#), este é o ponto fraco mais comuns em robôs pois para garantir sua segurança e confiabilidade é necessário que o mesmo tenha a capacidade de interpretar as variáveis ambientais. A percepção é o que torna os robôs diferentes de simples mecanismos, pois é ela quem dá a habilidade de adequar suas operações de acordo com as influências externas.

A percepção do ELIR pode ser definida como um sistema integrado de sensoramento e com unidades de processamento, em que seus dados serão utilizados como parâmetros de tomada de decisão e disponibilizados durante a operação de inspeção ao operador.

O sistema foi projetado de forma a possuir três subsistemas principais: segurança, georreferenciamento e detecção. A descrição de cada um dos subsistemas e suas funcionalidades serão mostradas nas próximas sessões.

Para um robô, o ambiente é um mar de ambiguidades, no qual ele vai afundar ou nadar a depender da robustez de sua percepção.

([FITZPATRICK, 2003](#))

### **2.1 *Estudo do estado da arte***

flkjaskldkfjaskldkfjs

### **2.2 *Descrição do sistema***

lasdjflsadjf

### 2.2.1 Especificação técnica

A construção do sistema de Percepção teve como base os requisitos técnicos do cliente. As especificações podem ser observadas abaixo.

- O sistema foi projetado para trabalhar com alimentação de 14V proveniente de baterias LiPo.
- A máxima temperatura de trabalho na *housing* é de 50 graus Celsius.
- O sistema consegue detectar objetos através do sonar em uma faixa de servidão de 6.45 metros.
- A obtenção de *frames* da câmera IR acontece na taxa de 1 frame a cada dois segundos.
- Em condições de sobretemperatura ou sobrecorrente o sistema alertará o operador.
- O sistema não é protegido contra ingresso de água

### 2.2.2 Arquitetura geral do sistema

lkasjdfklsdajflk;

### 2.2.3 Arquitetura de software

A arquitetura de software foi projetada em três camadas a fim de facilitar o desenvolvimento do sistema e simplificar o entendimento do mesmo. As camadas de são:

- *User Interface Layer*
- *Business Layer*
- *Drvier Layer*

As camadas e seus componentes podem ser vistos na Fig.2.1.

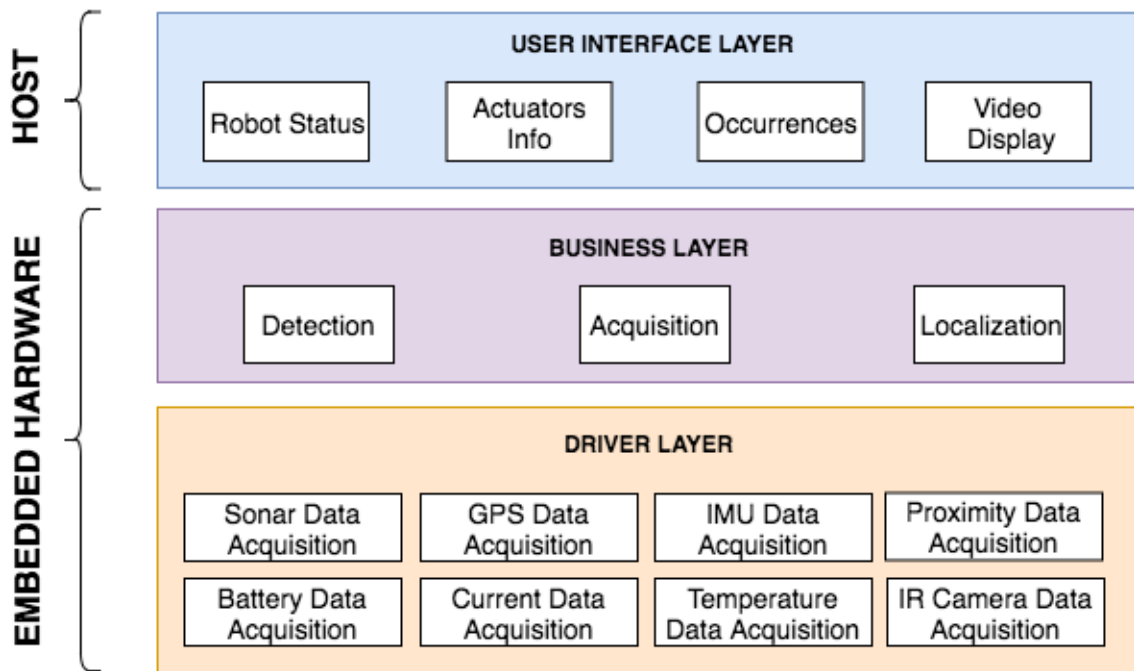


Figura 2.1: Arquitetura Geral da Perception

### 2.2.3.1 Driver Layer

A camada de *Driver Layer* está diretamente relacionada a funcionalidade de aquisição de dados. Ela composta pelo *hardware*, representado pelos sensores e seus respectivos drivers de comunicação. Desta forma, as subcamadas são nomeadas com o processo de aquisição de dados de cada sensor envolvido no projeto.

As subcamadas *Current Data Acquisition*, *Temperatura Data Acquisition*, *Proximity Data Acquisition* e *Sonar Data Acquisition* são responsáveis por adquirir as informações analógicas de seus sensores e transformá-los em dados da grandeza física a ser medida. Todas estas subcamadas utilizam a placa de interfaceamento Phidgets para o estabelecer de comunicação entre o computador (NUC) e os sensores.

As subcamadas *IMU Data Acquisition* e *GPS Data Acquisition* são responsáveis pelo recebimento de dados da IMU e do GPS seguindo o protocolo de comunicação do fabricante. Esses dois módulos estão conectados ao HUB USB da placa de interfaceamento Phidgets.

A subcamada de *IR Camera Data Acquisition* é responsável pela aquisição de dados da câmera térmica, a qual se comunica via VOSPI com um microcontrolador de arquitetura ARM (STM32F401RE) e converte os dados para USB para a NUC. Por último, a subca-

mada de *Battery Data Acquisition* é responsável pelo estabelecimento da comunicação e coleta de informações com o *Smart Charger* de bateria utilizando protocolo SMBUS.

As conexões e diagramas elétricos podem ser vistos no apêndice C.

### 2.2.3.2 *Business Layer*

A camada *business layer* é responsável por implementar a regra de negócio do sistema. As funcionalidades do sistema são representadas como sub-camadas da business layer, pois são elas responsáveis pelo processamento e coordenação dos dados adquiridos pela camada de aquisição.

### 2.2.3.3 *User Interface Layer*

A camada de *User Interface* foi projetada para disponibilizar os dados para o operador. Nela será mostrado de forma resumida os dados mais relevantes do robô e da operação. Nesta camada existem três subcamadas: *Robot Status Display*, *Actuators Display* e *Video Display*.

A subcamada *Robot Status Display* disponibiliza os dados de integridade do robô como temperatura, corrente, tensão, nível de bateria, entre outras informações. A subcamada de *Actuators Display* disponibiliza o dados de todos os motores do robô, como carga, temperatura, status e corrente. Por último, a subcamada de *Video Display* mostra em tempo real o monitoramento realizado pela câmera térmica, possibilitando o usuário ver os componentes da linha que estão com temperatura elevada e até mesmo identificar pontos quentes.

A interface irá se resumir em duas telas: A tela principal com um layout de *dashboard*, e outra que terá as informações dos atuadores. O *dashboard* será um painel de monitoramento, no qual haverá as informações mais importantes da missão, como pode ser visto na Fig. ?? no apêndice D. Essa tela irá mostrar as informações de integridade do robô, ocorrências e a imagem térmica. A tela dos atuadores irá mostrar de forma organizada, as informações já mencionadas, além da corrente total de cada HUB de motores. Pode-se observar a tela de atuadores na Fig. ?? no apêndice D.

## **2.3    *Desdobramento da função qualidade***

asdfsda<sup>sf</sup>

### *2.3.1    Requisitos técnicos*

asdfsad<sup>fsd</sup>

---

## Materiais e Métodos

---

asdfasdfsdf

### **3.1 Especificação dos componentes**

asjdfkldjsaf

#### *3.1.1 Estrutura analítica do protótipo*

asdkjfsdalkjf

#### *3.1.2 Lista de componentes*

asfkjdsahfkjs

### **3.2 Diagramas mecânicos**

asdfsdaf

### **3.3 Modelo esquemático de alimentação e comunicação**

asdfadsfsdfs

#### *3.3.1 Diagramas elétricos*

asdfsdaf

### 3.3.2 Esquemas eletrônicos

asdfsda

## 3.4 Especificação das funcionalidades

asdfsdfsdfs

### 3.4.1 Fluxo das informações

asdfsaf

### 3.4.2 Motion Planning

#### 3.4.2.1 Definição da funcionalidade

A funcionalidade de *Motion Planning* é responsável por realizar o planejamento da trajetória do Robô, utilizando o software *MoveIt!* que realiza o cálculo da cinemática inversa para encontrar a melhor forma de ultrapassar os obstáculos.

#### 3.4.2.2 Dependências

O software *moveit* pode utilizar o modelo matemático da cinemática inversa do robô ou um arquivo do tipo URDF. O nome URDF é uma sigla para *Unified Robot Description Format*, esse arquivo é uma especificação em XML utilizada para descrever robôs. Modelos em URDF apresentam uma simplicidade na descrição do robô, e para o caso do Robô *Elir*, utilizar o modelo URDF possibilitará uma aproximação fiel ao modelo real do robô, assim para o cálculo da cinemática inversa será utilizado o seu modelo URDF e não o seu modelo matemático.

#### 3.4.2.3 Premissas Necessárias

Para o correto funcionamento dessa funcionalidade as seguintes premissas são necessárias:

- A configuração dos limites de giro das juntas do robô estarão compatíveis com os comandos enviados
- O modelo URDF do robô estará adequado com o modelo físico
- O pacote gerado pelo *MoveIt! Setup Assistant* estará configurado adequadamente

#### 3.4.2.4 Descrição da Funcionalidade

A movimentação do robô na linha acontecerá por movimentos de translação e transposição de obstáculos. A translação na linha será feita por controladores de torque nas rodas do robô, enquanto a transposição dos obstáculos utilizará o moveit. Por meio da ferramenta *MoveIt! Setup Assistant*, se utiliza o modelo do robô para criar um pacote do ROS com os principais arquivos pelo moveit. A configuração correta do moveit possibilita que se utilizem as funções da sua biblioteca para o cálculo da trajetória, levando em consideração também obstáculos no caminho.

O moveit fornece uma *user interface* que recebe o end-effector, a nomenclatura atribuída ao node feito em python que recebe o *end-effector* é `moveit_commander`. O *node* responsável por fazer a integração da user interface com os parâmetros recebidos pelo *ROS Parameter Server* com o *end-effector* para fazer os cálculos é denominado `move_group`. O *node* `move_group` também pode receber parâmetros como leituras dos sensores do robô e nuvens de pontos.



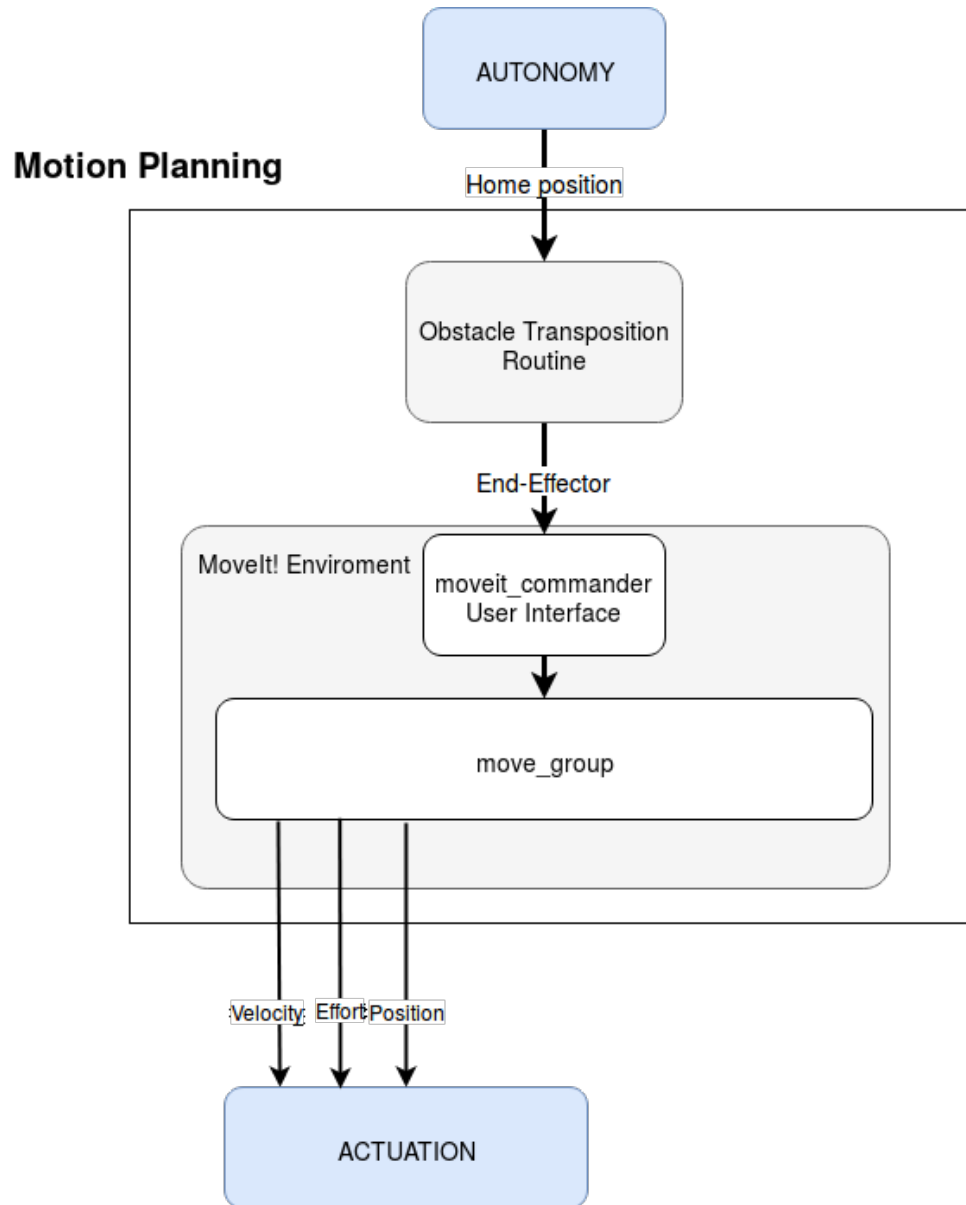


Figura 3.1: Fluxograma de funcionamento da funcionalidade de Motion Planning

Fonte: Própria

#### 3.4.2.5 Saídas

Por meio da compatibilização do *MoveIt!* com o *ROS*, a saída dessa funcionalidade são os comandos de velocidade, esforço e posição para cada junta do robô.

### 3.4.3 Actuation

#### 3.4.3.1 Definição da funcionalidade

A funcionalidade de Actuation tem como objetivo mover a estrutura física do robô, possibilitando o controle dos movimentos das juntas, garras e unidades de tração.

#### 3.4.3.2 Dependências

Essa funcionalidade depende das funcionalidades de *Power Management* e *Motion Planning*. O *Power Management* será responsável por fazer alimentação dos motores, possibilitando controlar a corrente máxima fornecida para cada grupo. A dependência em relação à funcionalidade de *Motion Planning* está atrelada principalmente com o software *MoveIt!*, que ao receber um *end-effector*, realiza o cálculo de trajetória e envia os comandos de velocidade, esforço e posição para os controladores das juntas, garras e unidades de tração.

#### 3.4.3.3 Premissas Necessárias

Para o correto funcionamento desse módulo, devem ser consideradas as seguintes premissas:

- Os motores devem estar configurados de acordo com o padrão de ID determinado pela equipe, fazendo parte da mesma malha de controle;
- Os controladores das juntas, garras e unidades devem estar configurados de acordo com os comandos que serão recebidos pelo *MoveIt!*;
- Os 3 grupos de motores estarão em malhas de alimentação de 12V individuais.

#### 3.4.3.4 Descrição da Funcionalidade

O ROS disponibiliza uma série de drivers para compatibilização dos motores dynamixel, possibilitando a criação de controladores específicos no seu ambiente. Serão criados os controladores referentes as juntas e unidades de tração do robô. Os controladores receberão comandos de *velocity* e *position* do *MoveIt!* junto com os comandos para movimentar o

robô na linha. Após os comandos serem recebidos pelos controladores, eles serão enviados para o *hardware* do robô, de acordo do padrão de comunicação dos motores, por meio de comunicação serial.

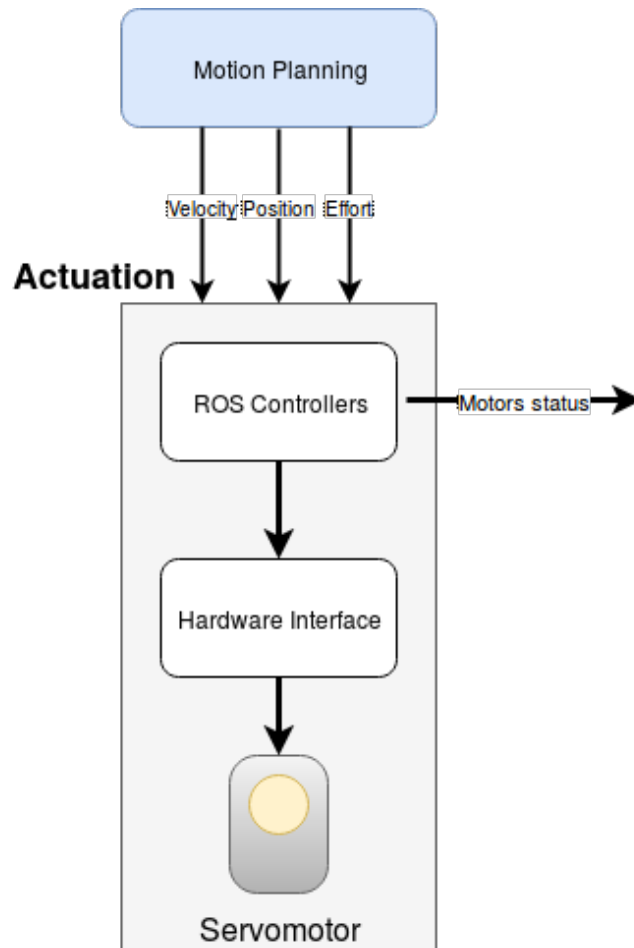


Figura 3.2: Fluxograma da funcionalidade Actuation

Fonte: Própria

#### 3.4.3.5 Saídas

A saída desta funcionalidade é o movimento da estrutura física do robô, que estará de acordo com o planejamento de trajetória do *MoveIt!* e com as instruções para operação na linha

### 3.4.4 *Power Management*

#### 3.4.4.1 *Definição da funcionalidade*

A funcionalidade de *Power Management* é responsável por administrar o fornecimento de energia para os dispositivos eletrônicos do robô, nos níveis adequados de tensão e corrente.

#### 3.4.4.2 *Dependências*

Essa funcionalidade depende da comunicação serial por meio da biblioteca *rosserial* e da operacionalização do firmware embarcado no hardware (placa) de acordo com as necessidades do projeto.

#### 3.4.4.3 *Premissas Necessárias*

Para o correto funcionamento desse módulo de *Power Management*, devem ser consideradas as seguintes premissas:

- A placa multiplexadora estará conectada diretamente ao módulo de *Power Management*
- Todos os dispositivos estarão conectados nas suas respectivas entradas
- A placa deverá ser alimentada por 2 baterias
- A placa estará conectada diretamente na NUC, por meio de uma USB

#### 3.4.4.4 *Descrição da Funcionalidade*

A placa de *Power Management* fornece diversos recursos para integração com o ROS. Seu firmware, além de realizar as medições e controle dos níveis de tensão e corrente para alimentação do robô, estará adaptado com as seguintes funcionalidades para que haja integração do hardware com o ROS:

- *Publishers* que contém os status das portas em níveis de tensão e corrente; avisos de surtos de corrente ou sobre-corrente; disponibilidade do hardware de *Power Management*

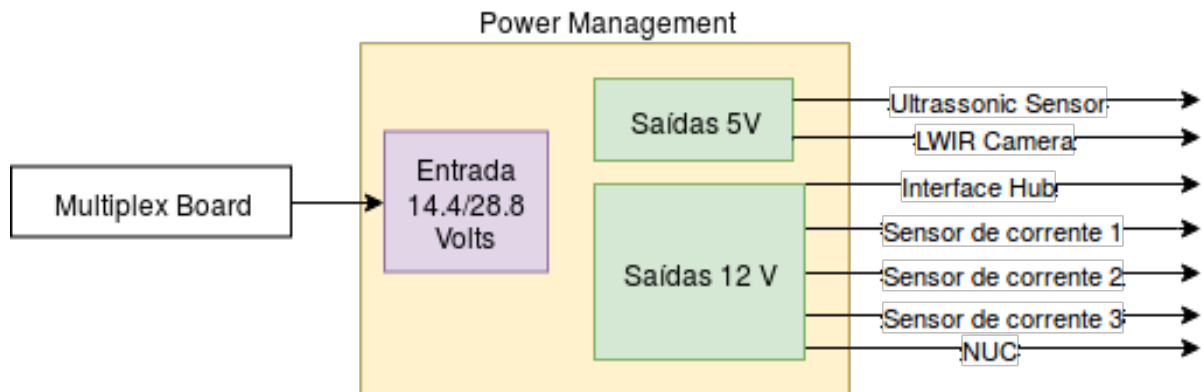


Figura 3.3: Fluxograma de funcionamento da funcionalidade de Power Management

Fonte: Própria

- *Serviços* para realizar a verificação dos níveis de corrente; definição dos limites de corrente nas portas; realização de comandos on-off

O conjunto de baterias fornecerá a energia para o sistema, a placa de *Power Management* irá administrar a distribuição da energia para os seguintes componentes:

- Grupos de servo motores
- Grupo de sensores de corrente
- NUC
- Interface HUB
- Câmera LWIR
- Sensor ultrassônico

#### 3.4.4.5 Saídas

A funcionalidade irá disponibilizar a energia para o robô e as seguintes estruturas no ambiente ROS:

- Tópicos com informações de tensão e corrente nas portas
- Tópico para aviso de sobre-corrente
- Tópico para informar disponibilidade da placa
- Serviços para ler e configurar limite de corrente das portas
- Serviço para ligar ou desligar energia em uma porta

### 3.4.5 *System Integrity Check*

#### 3.4.5.1 *Definição da funcionalidade*

É a funcionalidade responsável por checar a integridade do sistema antes do início da missão, verificando os subsistemas e suas variáveis.

#### 3.4.5.2 *Dependências*

A funcionalidade receberá informações dos seguintes componentes

- Sensor de Temperatura
- Servomotores
- Câmera IR
- Câmera Stéreo
- IMU
- Sensor de Proximidade
- Placa de Power Management
- Sonar
- Baterias

Todas as informações serão enviadas por meio do ambiente ROS, na forma de *Services* ou *Publishers*.

#### 3.4.5.3 *Premissas Necessárias*

As premissas necessárias para o funcionamento dessa funcionalidade são:

- Os subsistemas do robô irão disponibilizar o seu status no ambiente ROS por meio de tópicos ou serviços
- A checagem fará parte do planejamento de missão

#### 3.4.5.4 Descrição da Funcionalidade

A checagem da integridade do sistema é uma funcionalidade essencial para garantir o sucesso da missão e preservar a integridade do robô. O ROS facilita essa comunicação entre os subsistemas, possibilitando que seja criada uma rotina de checagem antes de cada missão.

Será disponibilizado no sistema uma rotina para iniciar a missão. Ao receber o comando para início de missão, os sistemas serão checados sequencialmente, utilizando estrutura de *Services* e *Publishers* do ROS. Caso algum sistema apresente falha, a missão não se iniciará e o erro será mostrado no *terminal* e registrado no arquivo de *log*. Se todos os sistemas estiverem em funcionamento, se iniciará a missão. O fluxograma da funcionalidade está ilustrado na figura 3.4.

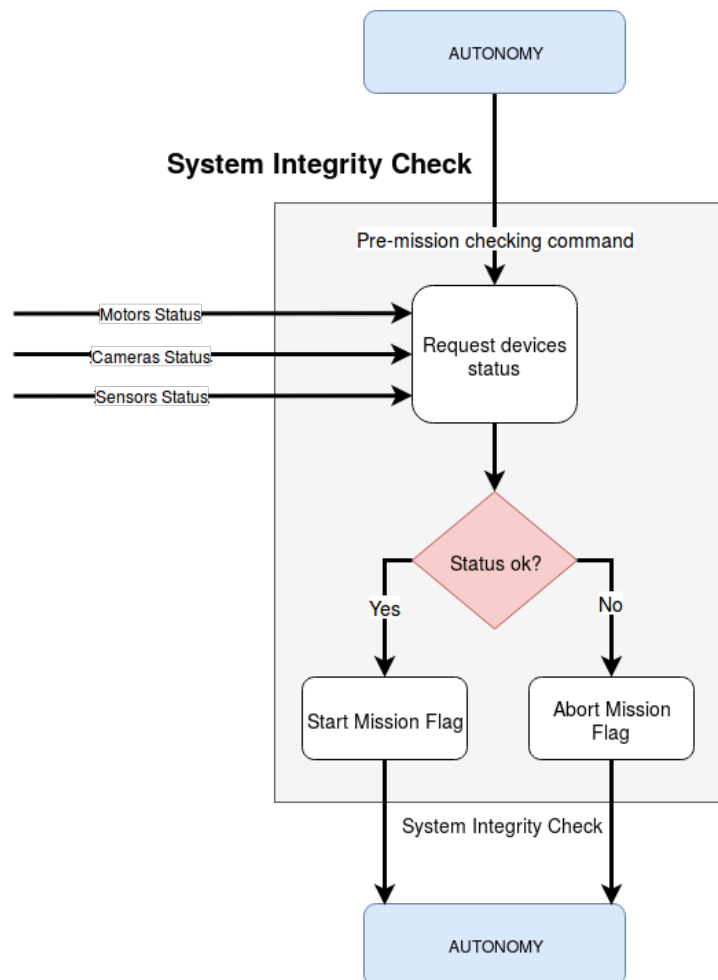


Figura 3.4: Fluxograma da rotina para checagem do sistema

Fonte: Própria

#### *3.4.5.5 Saídas*

No início da rotina de inspeção, a funcionalidade será responsável por enviar o sinal inicia a missão. Caso todos os sistemas checados estejam funcionando, a inspeção ocorrerá normalmente, se algum sistema apresentar defeitos, o defeito será mostrado no *terminal*, registrado em log e a missão será abortada.

### **3.5 Interface do Usuário**

asdfadsfsdfs

### **3.6 Simulação do sistema**

asdfadsfsdfs



---

## Resultados

---

asdfsdfsdf

### ***4.1 Testes unitários***

asdfsdfsdfs

### ***4.2 Testes integrados***

asdfsdfsdfs

### ***4.3 Avaliação da prontidão tecnológica***

asdfsdfsdfs

### ***4.4 Trabalhos futuros***

asdfsdfsdfs

---

## Conclusão

---

Chegou a hora de apresentar o apanhado geral sobre o trabalho de pesquisa feito, no qual são sintetizadas uma série de reflexões sobre a metodologia usada, sobre os achados e resultados obtidos, sobre a confirmação ou rechaço da hipótese estabelecida e sobre outros aspectos da pesquisa que são importantes para validar o trabalho. Recomenda-se não citar outros autores, pois a conclusão é do pesquisador. Porém, caso necessário, convém citá-lo(s) nesta parte e não na seção seguinte chamada **Conclusões**.

### **5.1 Considerações finais**

Brevemente comentada no texto acima, nesta seção o pesquisador (i.e. autor principal do trabalho científico) deve apresentar sua opinião com respeito à pesquisa e suas implicações. Descrever os impactos (i.e. tecnológicos, sociais, econômicos, culturais, ambientais, políticos, etc.) que a pesquisa causa. Não se recomenda citar outros autores.

---

**QFD**

---

---

## Diagramas mecânicos

---

## Diagramas eletro-eletrônicos

---

## Wireframes

---

---

## Logbook

---

---

## Referências Bibliográficas

---

FITZPATRICK, P. *Perception and Perspective in Robotics*. Cambridge, MA, 2003. [2](#)

MICHAELIS. *Percepção*. 2004. URL: <http://michaelis.uol.com.br>. [2](#)

THORPE, C. et al. *Dependable Perception for Robots*. [S.l.], 2003. [2](#)



*Desenvolvimento do robô de inspeção.*

Michael Faraday

John Nash

James Clerk Maxwell

Nikola Tesla

Salvador, Setembro de 2018.